Function Overloading

- Function overloading is calling two functions of same name with different parameters in same program.
- The parameters can be “datatypes”, “variables” & “number of variables” also.
- In the given code below we have 3 functions of same name “add” but with different parameters i.e in first function we have number of parameters= 2 (int x, int y).
- Similarly in second function we have number of parameters=3 (int x, int y, int z).
- And in the third function we have number of parameters = 2 but they are having Float as datatype.

```
Int add (int x, int y)
{
return x+y;
}

Int add (int x, int y, int z)
{
return x,y,z;
}

Float add (float x, float y)
{
return x+y;
}

Void main ()
{
 Int a = 10, b=5,c,d;
c=add(a,b);
d=add (a,b,c);
```

```
float i=2.5, j=3.5, k;
k=add (i, j);
}
```

- C++ compiler is capable enough to differentiate the same name functions accordingly and the compiler execute the correct function of its requirments.

- By function overloading we don't have to remember too many functions name while doing large programs simply we can create functions of same name with different parameters and it will be executed at its time of requirements.

(note: return type is never considered in a function overloading. If the functions name and the parameters are exactly same but return type is different that means they are not overloaded, they are having name conflicts)

Example :

```
Int max ( int x, int y)
Float max (int x, int y)
```